# MkTimeUSA Manual (FourthDraft)

Kaice T. Reilly

March 13, 2002



## 1 What MkTimeUSA(mktimeusa) Does

The mktimeusa script executes the MkTimeUSA code and a variety of ftools on an entire directory of USA event observations. The directory must contain an event file, a housekeeping file and a background file (created by usabckgnd) for each observation. The files should have the suffixes evt.fits, hkp.fits and bck.fits, respectively.

MkTimeUSA is a code which generates a set of good time intervals(gtis) for an observation. These gtis are based on housekeeping and event data. Once these gtis are created, ftools are used to plot key data in postscript files, cut out regions of bad data as delimited by the MkTimeUSA gtis and merge the MkTimeUSA gtis with the gtis contained in the original event fits files. The end product is a .filt file which contains only good data and a corresponding gti table.

## 2 How to Run MkTimeUSA

The general command for running mktimeusa is

mktimeusa your_data_dir output_dir bsub=[0 | 1] (nospacesaround'=')

where, your_data_dir is the path to the un-cut fits file directory and output_dir is the path to the output directory, output_dir can be ./, and bsub is for batch jobs (SLAC). Full paths are not necessary but they are safer. At SLAC, try not to use /a paths use /nfs paths instead.

If you want to use the mktimeusa default cuts then all you need is a directory containing your data. Then type

mktimeusa your_data_dir output_dir bsub=0

This is assuming that the job will be run in non batch mode. The output_dir need not exist although it can.

If you want to use your own cuts then you must first create output_dir and copy $USAroot/scripts/mktimeusa.in to output_dir

cp $USAroot/scripts/mktimeusa.in output_dir

Make any desired changes to the arguments in mktimeusa.in and then issue the following command:

mktimeusa your_data_dir output_dir bsub=0.

Many files will be created in the directory in which you start running the job. So you should not start several jobs from the same directory. This could cause problems. This is especially true when running in batch mode. In batch mode no files are moved to the output_dir. All output files are created in the directory where the user starts the job. When all batch jobs are finished the user should execute the command:

sh cleanup.sh

and this will put all of the files where they belong.

# 3 MkTimeUSA Argument Descriptions

Here is a description of the cut parameters used in MkTimeUSA. The default input for these values is shown in mktimeusa.in. mktimeusa.in is located in $USAroot/scripts. These defaults will probably change as we learn more about what works.

**MinimumHV** This is the minimum acceptable high voltage in the chamber. If a section of data has a high voltage less than MinimumHV then it will be cut out.

**PoleRatioMultiplier** is multiplied by the RatioMultiplier(below). Because the polar regions are expected to have higher electron rates, this parameter is used to make the electron cut less sensitive in those regions.

**MaximumOffset** is the maximum allowable offset(in degrees) from directly pointing at the source. Note the offset is given in the HKP file as calculated using the routine (name of new point goodroll routine) from usalib.

**MinimumOffset** is the minimum allowable offset(in degrees) from directly pointing at the source. Note the offset is given in the HKP file as

calculated using the routine (name of new point goodroll routine) from usalib. This quantity is intended for use in finding the off source segments in an observation.

**MinimumLimbAngle** is the minimum allowable angle(in degrees) between the pointing direction of USA and the Earth's limb. 5.0 degrees appears to work very well. Changing this parameter to smaller values will tend to contaminate an observation with x-rays coming off of the Earth's limb.

**RatioMultiplier** is a constant which multiplies the "ratio" of the top layer channel 15 (or 7) to the bottom layer channel 15 (or 7). The "ratio" is a number we came up with at SLAC based on rough estimates of the energy calibration. This ratio is set to 1.5 in MkTimeUSA. The value of 1.5 is based on the layer 0 and layer 1 efficiencies. Unfortunately, 1.5 is too strict of a cut so RatioMultiplier is used to tune the ratio. Large values of this parameter will allow regions with high rates of soft electrons to remain in the data.

**FracAboveBckgnd** is the acceptable fraction of veto counts above a calculated low point. MkTimeUSA goes through each observation and determines the minimum background veto level. Using FracAboveBckgnd MkTimeUSA cuts out regions where *any* veto rate is greater than (1+FracAboveBckgnd)*(The minimum calculated background).

**MaximumCh15Diff** is another parameter for handling the soft electrons, discussed above, in a slightly different way. This parameter simply sets a limit on the acceptable difference between the top and bottom layers in channel 15 (or 7). Greater differences allow more soft electrons.

**TimeBeforeOnSource** is the number of seconds after we are pointing at the source(i.e.. there is an acceptable offset) that we wish to start considering the data as good. This is mainly used to avoid the rate spike that seems to occur in all USA data at the beginning of an observation. It can avoid other problems.

**NbinsAVG** is the number of bins, one second, to average. This value sets the number of bins to average when determining the minimum calculated background. In most cases the more bins averaged together the higher the minimum calculated background rate will be.

**MinDataLength** the sets the minimum number of contiguous seconds of good data for that entire segment of data to be kept. If MinDataLength is 4 and a files longest uninterrupted segment of data is 3 seconds then no good data will be obtained from that file.

**TimeBeforeOffSource** is the opposite of TimeBeforeOnSource. This is the number of seconds prior to the last event frame stop to stop considering

any data as good.

**MinimumYawAngle** is self descriptive. This is a parameter used to cut out observations which are heavily shaded by the yoke. When yoke obscuration is very large it is, at this moment, difficult to accurately correct the counting rate.

**MinPVtoTCVRatio** is the minimum ratio of perimeter veto counts to total coincident veto counts. This is another way of cutting out regions containing large amounts of soft electrons. A value of 0.58 should cut out most soft electron regions.

**RateMultiplier** is multiplied by the average number of counts in channel 15 (or 7) of the top layer. If the current second has a count rate greater than this product then the current second is excluded. For better worse this quantity will exclude regions in which the count rate spikes. Making this number very large will include data spikes if they exist.

**PoleMaxCh15DiffMultiplier** is multiplied by **MaximumCh15Diff** in order to account for higher occurrence of soft electrons at the poles.

# 4 MkTimeUSA Output

The primary output of mktimeusa is a filtered event and background file, located in a subdirectory of the output directory called **FILT**, for each observation. These files will have a .filt suffix and contain only data in regions which pass the user's specified criteria for good data. The criteria used are either the defaults contained in the MkTimeUSA code or they are set in a file called mktimeusa.in which should be located in the desired output directory(see below).

If the user wants to know more about what mktimeusa does, they may look in the **SH** subdirectory. This directory contains a shell script for each observation. Here you will see all commands executed to give the data products. The output of these commands is stored in a file in the **LOG** subdirectory. There should be one log file for each evt.fits file. The vast majority of this output represents each time a second of data is lost due to cuts. In general, each time a second is lost a line is printed out. The first three characters on this line are either HKP or EVT. HKP indicates that the second was lost to some quantity in the HKP data(i.e. the offset). EVT indicates that something derived from the event data(i.e. channel 15 (or 7) count difference) caused the second to be lost. The logfiles are critical in understanding what mktimeusa does. *If ever seems that the data filtering is not correct then the first place to look for a problem is the* **LOG** *directory.*

In the subdirectories **EVTps** and **HKPps** mktimesusa gives multiple plots on one second intervals for each observation. For the event data three postscript files are generated: evt1.ps evt2.ps and evt3.ps are the suffixes. The first file shows data after cuts are made and contains a plot of the total count rate, yoke corrected total count rate, collimator response and the pointing offset versus time. The second file contains is the same as the first except it shows the unfiltered data. The third event postscript file plots some derived quantities, used to make selections of good data, from the unfiltered event data. These are the channel 15 difference, the channel 15 ratio, the total difference and the total ratio between layer 0 and layer 1.

In the **HKPps** subdirectory there are six postscript files for each observation: similarly hkp1-6.ps are the suffixes. The plots in these files cover all times, for which there are data, given in the housekeeping files. All quantities are plotted with respect to time. The first file contains the detector 1 layer 0 count rate, detector 1 layerl 1 count rate, the total coincident veto and the detector 1 perimeter veto. The second file contains the detector 1 layer 0 coincident veto, detectory 1 layer 1 coincident veto, the sae pitch steps and the collimator response. The third file contains the ratio of the perimeter veto to the total coincident veto, the yaw, the pointing offset and the limb angle. The fourth file contains the longitude of ARGOS, the latitude of ARGOS, the detector 1 electronic plate temperature and the detector 1 back plate tmperature. The fifth file contains the pointing RA, the pointing DEC, detector 1 high voltage for layer 0 and the detector 1 high voltage for layer 1. The sixth file contains the ARGOS eci x position, eci y position, eci z position and the USA telemetry mode.

The general purpose of the plots is to give the user a quick way of checking that the cuts they are making are reasonable. When MkTimeUSA is run several directories are created and the output is placed in its appropriate location. You should always have a look at these plots to make sure the cuts are what you think they should be. When you are done most of the files generated by mktimeusa may be deleted.

In addition to the postscript files contained in **EVTps** and **HKPps** there is the subdirectory **QDP_files**. Within this directory one will find and qdp command file (.pco) and data file (.qdp) for each of the postscript plots. Any of these files may be used to regenerate and toy with any of the postscript plots made. Further, all of the data and some derived parameters are there in an easily accessible ascii format.

In the output directory there should also be a **PHA** directory with a .pha file for each observation. The .pha files contain the normalized uncorrected energy spectrum for good data in each of the observations.

The **STAT** directory contians .stat files which contain a variety of information including count rates by channel and layer. However, the variety of information is so wide that the entire next section (see section 4.1) has been dedicated to the .stat files.

Two less important subdirectories are **EDIE** and **TIME** In **EDIE** are .edie files contain differential energy spectra of each observation (we will see if these are ever useful). The hope is that these might someday be useful for correcting the $400Hz$ power spectral feature (see section on EDIE). All files related to the good time interval construction are stored for each observation in the **TIME** directory. Clever use of ftools and these time files make it possible to re-filter data without re-running mktimeusa.

Finally there are three log files created in the top level of the output directory. The file logfile.log summarizes some of the outcome of mktimeusa. The files ObsLengthLog.log and ObsLengthLogRaw.log give the total length of each observation before filtering. They both contain the same information except that ObsLengthLog.log matches an observation name with the observation length and does a tally of unfiltered observation time. If there is a need for the number of remaining or good seconds then the following command can be use in the **STAT** directory:

$$\text{zcat } * \text{.stat.gz} \mid \text{grep } -\text{v HKP} \mid \text{wc } -\text{l}. \tag{1}$$

Now for the .stat files.

## 4.1 STAT File Contents

Perhaps the most useful output, other than the .filt file, is the file associated with each observation having the suffix .stat. These files contain a variety of information about all seconds in an observation which passed the cuts. Here are the column headings for the stat files.

```
1:HKPtime 2:EVTtime 3:BCKtime
4:Lyr0Chn0Rate 5:Lyr1Chn0Rate
6:Lyr0Chn1Rate 7:Lyr1Chn1Rate
8:Lyr0Chn2Rate 9:Lyr1Chn2Rate
10:Lyr0Chn3Rate 11:Lyr1Chn3Rate
12:Lyr0Chn4Rate 13:Lyr1Chn4Rate
14:Lyr0Chn5Rate 15:Lyr1Chn5Rate
16:Lyr0Chn6Rate 17:Lyr1Chn6Rate
18:Lyr0Chn7Rate 19:Lyr1Chn7Rate
20:Lyr0Chn8Rate 21:Lyr1Chn8Rate
```

```
22:Lyr0Chn9Rate 23:Lyr1Chn9Rate
24:Lyr0Chn10Rate 25:Lyr1Chn10Rate
26:Lyr0Chn11Rate 27:Lyr1Chn11Rate
28:Lyr0Chn12Rate 29:Lyr1Chn12Rate
30:Lyr0Chn13Rate 31:Lyr1Chn13Rate
32:Lyr0Chn14Rate 33:Lyr1Chn14Rate
34:Lyr0Chn15Rate 35:Lyr1Chn15Rate
36:TotalRate 37:ArgosLat 38:ArgosLong
39:Sae_Yaw 40:YawDeg 41:YokeObs
42:YokeCorrectedRate 43:det_1_pv
44:d1_ep_temp 45:d1_bp_temp
46:d1_l1_cv 47:d1_l2_cv 48:d1_tot_cv
49:CollimatorResponse 50:PointingOffset
51:limb_angle 52:Node
53:Lyr0Chn0Bck 54:Lyr1Chn0Bck
55:Lyr0Chn1Bck 56:Lyr1Chn1Bck
57:Lyr0Chn2Bck 58:Lyr1Chn2Bck
59:Lyr0Chn3Bck 60:Lyr1Chn3Bck
61:Lyr0Chn4Bck 62:Lyr1Chn4Bck
63:Lyr0Chn5Bck 64:Lyr1Chn5Bck
65:Lyr0Chn6Bck 66:Lyr1Chn6Bck
67:Lyr0Chn7Bck 68:Lyr1Chn7Bck
69:Lyr0Chn8Bck 70:Lyr1Chn8Bck
71:Lyr0Chn9Bck 72:Lyr1Chn9Bck
73:Lyr0Chn10Bck 74:Lyr1Chn10Bck
75:Lyr0Chn11Bck 76:Lyr1Chn11Bck
77:Lyr0Chn12Bck 78:Lyr1Chn12Bck
79:Lyr0Chn13Bck 80:Lyr1Chn13Bck
81:Lyr0Chn14Bck 82:Lyr1Chn14Bck
83:Lyr0Chn15Bck 84:Lyr1Chn15Bck
```

Column 52: Node indicates on which orbital segment the observation took place. The Node code is as follows: +2, North Pole; +1, Ascending Node; -1, Descending Node; -2, South Pole.

Anything may be added to this file if it is derivable from what is contained in the housekeeping and event files.

## 5 To Be Done

Add rate correction for events lost due to dead time(Ganya's equation in "The Recipe for Correcting for Dead-time". Currently, the STAT file only includes an overall corrected rate due to yoke observation and unfortunately it is incorrect because background is not subtracted from that.(subtract background, correct for dead-time, correct for yoke, in that orde—still not implemented)

Develop two standard sets of cuts. One set will be the cuts where the the background model is matched and the other set will allow us to keep the maximum "safe and reasonable" amount of data for use in power spectra.

**We are still working on a best set of cuts. I need everyones help on this. If anyone has cut large data sets by hand then could do cuts on the same data set with mktimeusa. The number of seconds found in both cases may be compared and the cuts tuned in mktimeusa so that we can construct a reasonable and systematic way of filtering the USA data.**

In the future stat files should contain corrected count rates.

Other.....